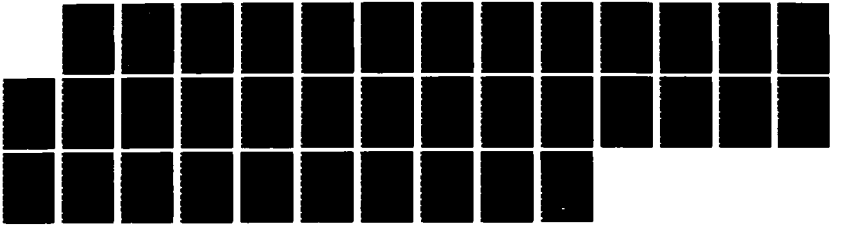
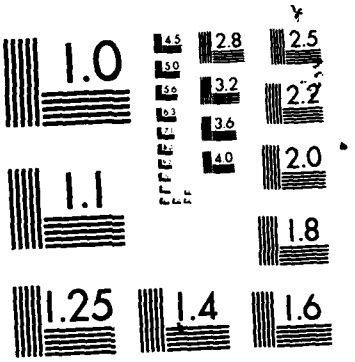
AD-A172 866 A FRAMEWORK FOR EMPIRICAL DISCOVERY(U) CALIFORNIA UNIV 1/1
IRVINE DEPT OF INFORMATION AND COMPUTER SCIENCE
P LANGLEY ET AL 24 SEP 86 UCI-ICS-TR-86-19
UNCLASSIFIED N00014-84-K-0345 F/G 9/2 NL





Information and Computer Science

A FRAMEWORK FOR EMPIRICAL DISCOVERY

Pat Langley
Bernd Nordhausen
Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine, CA 92717

TECHNICAL REPORT

UNIVERSITY OF CALIFORNIA
IRVINE

A FRAMEWORK FOR EMPIRICAL DISCOVERY

Pat Langley
Bernd Nordhausen
Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine, CA 92717

Technical Report 86-19

24 September 1986

We would like to thank Randy Jones, Don Rose, and John Gennari for their comments on an early draft of this paper and for discussions that led to the ideas presented therein.

This paper appears in the *Proceedings of the International Meeting on Advances in Learning*, Les Arcs, Frances, 1986.

This work was supported in part by Contract *N00014-84-K-0345* from the Information Sciences Division, Office of Naval Research.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 8	2. GOVT ACCESSION NO. AD-A172866	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A Framework for Empirical Discovery		5. TYPE OF REPORT & PERIOD COVERED Interim Report 5/86–8/86
		6. PERFORMING ORG. REPORT NUMBER UCI–ICS Technical Report 86–19
7. AUTHOR*(s)* Pat Langley Bernd Nordhausen		8. CONTRACT OR GRANT NUMBER*(s)* N00014–84–K–0345
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Information & Computer Science University of California, Irvine, CA 92717		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Information Sciences Division Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE 24 September 1986 13. NUMBER OF PAGES 31
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

This paper appears in the *Proceedings of the International Meeting on Advances in Learning*, Les Arcs, France, July 1986.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

machine learning
conceptual clustering
defining new terms
scientific discovery
chunking

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

OVER

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. ABSTRACT

Previous research in machine learning has viewed the process of empirical discovery as search through a space of 'theoretical' terms. In this paper, we propose a problem space for empirical discovery, specifying six complementary operators for defining new terms that ease the statement of empirical laws. The six types of terms include: numeric attributes (such as PV/T); intrinsic properties (such as mass); composite objects (such as pairs of colliding balls); classes of objects (such as acids and alkalis); composite relations (such as chemical reactions); and classes of relations (such as combustion/oxidation). We review existing machine discovery systems in light of this framework, examining which parts of the problem space were covered by these systems. Finally, we outline an integrated discovery system (IDS) we are constructing that includes all six of the operators and which should be able to discover a broad range of empirical laws.

1. Introduction

One approach to classifying learning tasks focuses on the independence of the learner. At one end of this spectrum lies learning from instruction and learning from examples, in which a tutor directs the learner's attention. At the other extreme lies learning by discovery, in which the agent is left to his own devices and must resort to observing the surrounding environment. Scientific discovery is an important instance of such 'observational' learning, and in this paper we focus on computational approaches to this problem.

Of course, scientific discovery is a varied and complex process that we cannot hope to cover here. Instead, we will devote our attention to *empirical* discovery in preference to other aspects of discovery, such as theory formation. The dividing line between these two behaviors is vague, but the basic distinction is clear. Empirical discovery leads to statements (laws) that *summarize* data, while theory formation produces statements (theories) that *explain* phenomena (often empirical laws). Thus, the ideal gas law, Kepler's laws, and the 'theory' of acids and bases are all empirical laws, since they simply describe observations. In contrast, the atomic theory, the kinetic theory of gases, and the caloric theory of heat are all theories, since one can use them to deduce various empirical laws.

For a number of related reasons, empirical discovery seems the natural place to begin developing models of the discovery process. Historically, such discoveries have preceded the development of scientific theories and in fact have provided the base for such theories. One might imagine general, weak methods for discovering empirical laws, while theory construction may require more domain-specific knowledge and thus more complex mechanisms. In addition, nearly all research in machine discovery has examined empirical laws, and we would like to build upon existing results rather than start anew.

In the following pages, we propose a framework for understanding empirical discovery. This framework takes the form of six operators for defining new terms; taken together, these operators define a problem space for empirical discovery. After presenting the basic scheme, we apply the framework towards achieving two goals. The first is to improve our understanding of earlier machine discovery systems, including the relations between these systems. Our approach here is to redescribe each systems in terms of the operators they employ. The second goal is to construct an integrated discovery system that deals with all of the major issues that arise in empirical discovery. In this case, we outline our plans for a system that invokes all six operators, along with some heuristics for selecting them.

2. A Problem Space for Empirical Discovery

Previous AI research on empirical discovery has viewed this process as heuristic search, and our framework will follow similar lines. One common feature of these earlier efforts is that they focused on constructing new terms to simplify the discovery of laws. For instance, Lenat's (1977) AM system spent much of its time defining terms such as natural numbers, multiplication, divisors-of, and prime numbers. These terms made the statement of laws (such as the unique factorization theorem) relatively easy. Similarly, BACON (Langley, 1981; Langley, Bradshaw, & Simon, 1983) spent the majority of its effort defining new terms like d^3/p^2 and *mass*. Once the appropriate terms had been defined, the system could state

its laws as simple constancies or linear relations.

Close examination of existing discovery systems and the history of science reveal distinct *classes* of defined terms. Some systems have focused on one subset of these classes, while other programs have focused on different subsets. Below we consider six different types of terms that may be constructed during the discovery process. In each case, we formulate an *operator* for defining that class of terms, and taken together these operators define a *problem space* for empirical discovery. Although we will not claim this space is exhaustive, we will see that it is more comprehensive than the spaces searched by earlier discovery systems. Below we discuss the operators in detail, presenting examples of each from the history of science.

2.1 Defining Numeric Terms

The most obvious operator for defining new terms involves numeric attributes, and this is the basis for both Langley et al.'s (1983) BACON system and Falkenhainer & Michalski's (1986) ABACUS system. Given observable numeric attributes $a_1, a_2, \ldots, a_n$ one can define a new numeric term $X = f(a_1, a_2, \ldots, a_n)$ which combines the observable attributes using mathematical functions such as multiplication or addition. For example, given attributes of a gas such as its *pressure* P, its *temperature* T, and its *volume* V, one can define a new term $X = PV/T$. Such terms may have a constant value or they may have simple relations to other numeric terms.

For instance, one statement of the ideal gas law is that the term PV/T is constant. But such terms are more than useful in stating empirical laws; they can also simplify the process of discovering such laws. Let us consider some examples from the history of science in which the definition of numeric terms aided the discovery process.

Different aspects of the ideal gas law were discovered in various forms by Avogardo, Boyle, Charles, and Gay-Lussac. These scientists detected constant relations between the volume of a gas, its temperature, and its pressure. Boyle noted that the volume V of a gas at constant temperature T is inversely proportional to the pressure P of the gas – that is, the term $X = PV$ has a constant value under constant temperature. Charles and Gay-Lussac observed (independently of each other) that the volume of gas at constant pressure is proportional to its absolute temperature. In other words, they discovered that the term $X = V/T$ does not vary, provided the pressure remains the same. Ultimately, these separate laws were combined into the more general ideal gas law.

Kepler's and Black's discoveries provide additional examples of numeric laws. The third law of planetary motion relates two observable attributes – the mean *distance* d of a planet from the sun and the *period* p of that planet. Kepler's statement of this law was that "the squares of the periods of revolution of the planets are proportional to the cubes of the mean distance to the sun." However, one can also state this law by defining the term $X = d^3/p^2$ and noting that the value of X is constant across all planets.

Black's heat law is more complex, involving two objects with different temperatures that are placed in contact. Over time, the temperature of one object increases and the other decreases until they become equal. The final temperature is a function of the initial temperatures, the masses of the objects, and the particular substances involved. This law

points out the need for our second class of terms – intrinsic properties.

2.2 Defining Intrinsic Properties

An *intrinsic property* is some term that, for a given object or class of objects, has a constant value over time. Thus, this value can be associated with the object/class and retrieved whenever that object/class is encountered. For instance, values of the intrinsic property *mass* are associated with specific objects, while values of the property *density* are associated with entire classes of objects (a 'substance'). Our second operator for empirical discovery is responsible for postulating intrinsic properties and inferring their values.

We denote an intrinsic property as $i\_p(O) = n$, where O is an object or object class and n its associated value for the intrinsic property $i\_p(O)$. Unlike numeric term such as PV/T, intrinsic properties cannot be directly defined in terms of observable attributes. Instead, they require some assumptions about the form of the law involved and the solution of simultaneous equations. However, once an intrinsic property has been defined and its values have been computed, it can be used in the same way as an observable attribute.

After the famous bathtub incident, Archimedes formulated the principle of displacement: the volume of a body immersed in fluid equals the volume of the liquid it displaces. Using this principle, Archimedes was able to measure the volume of an irregular object, and thus to determine its density and composition. This volumetric attribute can be viewed as an intrinsic property for which different irregular objects having different values. Once these values have been determined, they can be used to distinguish different objects from one another.

Mass is another intrinsic property that occurs in several quantitative laws, including conservation of momentum. For the collision of two objects, this law can be stated as:

$$m_1 v_1 + m_2 v_2 = m_1 v_1' + m_2 v_2'$$

where m_1 and m_2 are the masses of the two objects, v_1 and v_2 are the velocities before impact, and v_1' and v_2' are the velocities after impact. Given the form of this law and the ability to measure the velocities, we can determine the relative masses of the colliding objects. This involves solving simultaneous equations for the unknown masses, and this in turn requires enough equations to identify their values. If one wants to determine the masses of five different objects, then exactly five observed collisions are needed. Once the mass of an object has been identified, this value can be used in other experiments to discover still other laws.

2.3 Forming Composite Objects

The above operators focus on attributes, and such attributes must always be associated with a single object. However, the conservation of momentum law just described involves a constant relation *between* objects. One way to represent such relations involves defining a new *composite* object, and stating the law in terms of this composite's attributes. Given two or more objects O_1 and O_2, one can define a composite object O_c which has O_1 and O_2

as its components. We express this as:

$$O_c \equiv O_1 \ \& \ O_2$$

Such a composite object can be handled in the same way as an observable object, provided one can determine the values of its attributes. Many of these can be computed directly from the attributes of its component objects. For example, the mass of a composite object is simply the sum of the component masses, while the density involves a weighted average of the component densities.

In summary, composite objects are useful in stating empirical laws which relate some set of objects rather than describing a single object. Our third operator for empirical discovery is responsible for defining such composites. Such an action seems especially useful when a conservation law is involved. Let us consider the momentum example in more detail, in order to clarify the role of this operator and its interaction with the other operators.

The basic experimental situation involves two objects O_1 and O_2 that collide with each other. Based on the initial velocity v and the final velocity v' for each object, our second operator can define the intrinsic properties m (the mass of each object) and infer its value. Based on this property and the velocities, our first operator can define the numeric attributes $P = mv$ (initial momentum) and $Q = mv'$ (final momentum). No simple regularities arise from looking at these attributes for isolated objects. However, if one defines the composite object $O_c \equiv O_1 \ \& \ O_2$, and if one assumes that the momentum of O_c is the sum of its components' momenta, then the simple law $P_c/Q_c = 1$ emerges. This shows some of the *representational power one can achieve by defining composite objects.*

Now let us consider another example in which there is even more interaction between intrinsic properties and composite objects. If the surface of one body slides over the surface of another, the two bodies exert a frictional force on each other. The quantity of friction depends on the composition of the two objects and on the force pressing the bodies together, but is independent of the area of contact and the speed. This relationship can be expressed as

$$F_f = \mu F_n$$

where F_f is the frictional force, F_n is the normal force pressing the two objects together and μ is the friction coefficient.

The coefficient μ in the friction law can be viewed as an intrinsic property, but unlike most such properties, its values are a function of *both* substances. Thus, the friction coefficient for steel on steel is different than for aluminum on steel, and the best one can do is to store values with each pair of substances. Given this situation, it seems natural to define composite objects* such as steel-steel and aluminum-steel and to associate each intrinsic value with one such composite. In this way, we can retain the assumption that intrinsic values are associated with single objects, and leave the responsibility for creating such objects with our third operator.

---

* Actually, these are object *classes* rather than individual objects. Just as one can associate intrinsic values with classes of objects as well as specific objects, so can one form composites with object classes.

2.4 Defining Classes of Objects

Just as one can define composite objects, one can also define new *classes* of objects. Thus, one might decide that objects O_1, O_3, and O_7 have similar properties and belong to the same basic type, leading one to define a new group O_g with these three objects as members.* We will denote this new group as $O_g = \{O_1, O_3, O_7\}$. New terms of this form are quite useful in stating qualitative laws such as occurred in the early days of chemistry and biology. Furthermore, such groups can be modified incrementally; if one later encounters object O_{10} that is similar to existing members of the class O_g, then one may add O_{10} to the class. The process of defining classes can also be applied recursively to form a taxonomy or classification hierarchy. For instance, having defined the object classes O_g and O_h, one might group these together to define a higher level class O_m.

Such taxonomies aid the discovery of qualitative laws at different levels of abstraction. For example, early biologists spent much of their time defining different species, classes of species, and so forth. Similarly, the early chemists devoted considerable effort to defining classes such as alkalis, acids, and salts. In each case, these classes were defined not only by their members, but also by the features held in common by those members. These defining features can be viewed as qualitative empirical laws. Michalski (1980) has used the phrase *conceptual clustering* to refer to this task of formulating taxonomies and determining their associated descriptions.

Just as class formation can help in discovering empirical laws, so can the discovery of qualitative laws suggest new classes. For instance, Mendel experimented with self-fertilized peas and found that some yellow peas produced only yellow offspring, other yellow peas produced both yellow and green offspring, and green peas consistently had green offspring. Based on these observations, he defined the classes of *hybrids* and *purebreds* and formulated the laws of genetic segregation as follows:

$$\forall\ x \in \textit{purebreds}\ \ \textit{parent-of(x,y)}\ y \in \textit{purebreds}$$

$$\forall\ x \in \textit{hybrids}\ \ \textit{parent-of(x,y)}\ y \in \textit{hybrids} \vee y \in \textit{purebreds}$$

The first of these laws can be paraphrased 'All purebreds produce offspring which are purebreds.' The second empirical generalization can be restated 'All hybrids produce some offspring which are hybrids and some which are purebreds.' The two classes, together with the laws summarizing their behavior, formed the basis for the genetic theory.

As we have mentioned, classes may change their membership over time, and the details of this process may prove interesting. Early chemists first defined the classes of *acids*, *alkalis*, and *salts* in terms of their taste. However, they soon discovered that acids reacted with alkalis to form salts, and this empirical law gradually became a central feature of all three

* Note that the initial objects here are linked to the new object-class by an *instance-of* or *subset-of* relation. This contrasts with the *part-of* relations that holds between composite objects and their components.

classes. Ultimately, substances that did not taste sour were included as acids because they reacts with known alkalis to form salts. This shift also led to the more abstract class of *bases*; this included the subclasses of alkalis and metals, both of which reacted with acids to form salts.

2.5 Defining Composite Relations

Objects can be described by their attributes, but they can also be described through their *relations* to other objects, and this suggests a fifth type of defined term. Given a set of primitive relations between objects, one can define new composite relations. This is similar to the process of defining composite objects, except that one must handle the arguments of these relations.* For example, one can combine the relations *brother(X, Y)* and *spouse(X, Y)* to define the composite relation *brother-in-law(X,Z)*. This can be stated formally as:

$$\textit{brother-in-law}(X,Z) \Leftarrow \textit{brother}(X,Y)\ \&\ \textit{spouse}(Y,Z)$$

this means that X is the brother-in-law of Z if X is the brother of Y and Y is the spouse of Z. By composing an existing relation (such as *parent-of(X, Y)*) and a qualitative attribute (such as *color*), one can also define more specific relation (such as *parent-of-green-child(X, Y)*.

Similarly, one might define the inverse of an existing relation.

The definition of composite relations can be viewed as one form of *chunking*. Although the existing machine learning work on chunking (Neves & Anderson, 1981; Laird, Rosenbloom, & Newell, 1984) has focused on procedural knowledge, chunks can also be perceptually-oriented. In skill acquisition, chunking methods have been used to improve the problem solving process. In scientific discovery, the goal is instead to describe the behavior of objects and classes over time.

Many mathematical concepts can be viewed as relations defined in terms of simpler relations. For example, one can define multiplication in terms of the addition concept, and one can in turn use multiplication to define the concept *divisors-of*. This term can then be used in the definition of prime numbers, which are simply those natural numbers having only two divisors (themselves and one). Other concepts from number theory can be constructed along the same lines.

In a similar fashion, one can imagine Mendel defining the two restricted versions of the parent relation:

$$\begin{aligned}\textit{parent-of-green-child}(X,Y) &\Leftarrow \textit{parent}(X,Y)\ \&\ \textit{color}(Y,\textit{green})\\ \textit{parent-of-yellow-child}(X,Y) &\Leftarrow \textit{parent}(X,Y)\ \&\ \textit{color}(Y,\textit{yellow})\end{aligned}$$

Given these higher-level relations, one can more easily define the classes of purebred and hybrid peas. Purebreds consist of those peas satisfying only one of these relations, while the hybrid class contains those peas satisfying both relations.

---

* Forming composite relations is also similar to defining numeric attributes, but the latter take at most one object as their argument, while relations can take an arbitrary number. Also, the latter take on only numeric values, while relations describe qualitative links between objects.

2.6 Defining Classes of Relations

If one can define classes of objects, then one can define classes of *relations* as well, and our sixth operator is responsible for this process. Relational classes prove useful in that they can take the place of specific relations in the statement of qualitative laws. For instance, the electrical, magnetic, gravitational, and nuclear forces all differ in their details, but they have much in common as well. As a result, it makes sense to consider them as members of a more abstract force relation. Similarly, both the phlogiston and oxygen theorists held that combustion and rusting were instances of a related process, even though their superficial effects were different. Such relational classes let one state more general laws and make more predictions than a number of specific relations.

Let us consider an example of how this sixth operator might be used. Suppose that one does not yet have a general notion of *reactions*, but knows that when HCl and NaOH are combined, both substances disappear and a new substance NaCl appears (along with some water). Now suppose one combines HNO_3 and KOH, finding that a new substance KNO_3 appears (along with some water). After observing these two experiments, one might define the class of relations *alkali-combines-with-acid-to-form-salt* with these two specific relations as members. One can use this abstract relation in qualitative laws that describe object classes. Moreover, these laws can be used as 'data' in suggesting even more abstract relations, such as the general class of reactions.

The process of defining relational classes is the least well-explored of the operators we have described, and to our knowledge, none of the existing AI discovery systems have used this operator. As a result, it will not come into play during our review in the following section. However, we believe the process of defining relational classes is just as central to constructing a truly integrated discovery system as the other five operators we have discussed.

2.7 An Ordering on the Operators

In the previous sections, we described six operators for defining new terms which form a problem space for empirical discovery. Table 1 lists these operators and the formal notation we have introduced for each. For any reasonable domain, the search space which these operators define is extremely large. Therefore, a robust discovery system will require some *heuristics* to determine the best operator to apply in a given situation.

As we will see in the following section, existing AI discovery systems address only a subset of this problem space and use at most two of the operators. As a result, the problem of search control is not as serious for these systems.* A more complete response to this problem must take the form of an implemented discovery system which uses all of the operators, thus addressing the entire problem space and forcing a principled answer to search control. Yet a look at the history of science reveals an initial plausible ordering on the operators. Let us review the evolution of chemistry with this goal in mind.

---

* Actually, Lenat's (1977) AM has an agenda mechanism which lets the system select among tasks. Even though AM uses only two of our operators (defining composite relations and defining classes of objects), this agenda mechanism has the flavor of an integrated system.

Early chemists were concerned with the classification of chemical substances and with qualitative relations between these substances. This seems natural, since one must decide on a basic set of classes and relations before considering quantitative laws. They formed object classes such as *acids* and *alkalis*, originally defined in terms of simple qualitative attributes but eventually incorporating relational laws. They formed composite relations such as *acid-reacts-with-alkali*, and they also formed abstract classes of such relations. One of the early chemical controversies revolved around whether reactions and mixtures involved two different processes; this can be viewed as a debate about the appropriate classes of relations. Thus, three of our operators – forming object classes, defining composite relations, and forming relational classes – are employed early in the empirical discovery process.

Table 1. Operators and notation

OPERATOR	*NOTATION*
numeric term	$X = f(a_1, a_3, a_5)$
intrinsic property	$i\_p(O) = n$
composite object	$O_c \equiv O_1 \& O_4$
class of objects	$C_2 = \{O_1,\ O_3,\ O_7\}$
composite relation	$R(O_1, O_2) \Leftarrow R_1(O_1, O_2) \& R_2(O_1, O_2)$
class of relation	$R_c(O_1, O_2) = \{R_1(O_1, O_2),\ R_2(O_1, O_2),\ R_7(O_1, O_2)\}$

At the end of the 18th century, chemists shifted their attention from qualitative laws to quantitative aspects of chemical reactions. They stopped focusing on symbolic attributes such as color and taste,* and turned to numeric attributes such as volume and weight. This paradigm shift led directly to principles such as the conservation of mass, Proust's law of constant proportions, Dalton's law of simple proportions, and Gay-Lussac's law of combining volumes. These numeric laws related the masses and volumes of the substances involved in reactions, and all were discovered during the late 1700's and early 1800's.

* It is important to note that qualitative information was not abandoned when chemistry entered its quantitative stage. Qualitative features were still used to identify substances, and such identification was absolutely necessary to successful quantitative studies. However, such identification had become trivial at this point, and the major efforts of chemists were devoted to numeric aspects.

Upon closer examination, we find that the remaining three operators have a central role to play in these quantitative discoveries. For instance, suppose we observe the weight W_E of an element entering a reaction and the weight W_C of the compound that results. From these two terms, one can define the ratio W_E/W_C, and this numeric term has a constant value for any pair of substances. This is one version of Proust's law of constant proportions. Given such a constant value, it makes sense to define an intrinsic property and to associate it with the pairs involved for future use. However, the value is conditional on both the element and the resulting compound, so that we must first define a composite object and associate the intrinsic value with it. Similar interactions between these three operators occur for Dalton's and Gay-Lussac's laws, and the operator for defining composite objects also proves useful for stating conservation of mass.

To summarize, operators which promote qualitative discoveries (defining classes of objects, composite relations, and relational classes) generally precede operators which promote quantitative discoveries (defining numeric terms, intrinsic properties, and composite objects). However, the ordering on our operators is not as simple as we have suggested. Ultimately, these quantitative discoveries led to higher level 'data' which chemists used to formulate higher level classes. In particular, estimates of the intrinsic property atomic weight* led Mendeleev to propose his periodic table, which classified elements using two complementary taxonomies (corresponding to the rows and columns of the table). Hence, qualitative discoveries lay the foundation for quantitative discoveries, but the latter can in turn lead to still higher level qualitative laws.

3. Previous Research on Machine Discovery

Now that we have presented a problem space for empirical discovery, let us review some earlier research in this light. Below we review five existing discovery systems. In each case, we begin with an overview of the system. We then consider which of the operators that system employs to discover empirical laws, and examine the conditions under which it applies those operators. We will find that the existing systems search only a small part of the overall space we have defined, never using more than two of the six operators.

3.1 AM

Lenat (1977, 1978, 1982) carried out some of the earliest and best-known research on machine discovery, so it seems appropriate to begin our review by examining his AM system. The program begins with a set of some 125 concepts from elementary mathematics, such as 'set', 'ordered pairs', and 'equality'. Using these as its base, AM defines new concepts in terms of existing ones, arriving at familiar mathematical concepts such as 'natural numbers', 'addition', 'multiplication', and 'prime numbers'. The system also generates hypotheses that relate these concepts to each other, including the unique factorization theorem and Goldbach's conjecture.

* Actually, qualitative features also played an important role in Mendeleev's discovery, but atomic weight was a central component.

AM represents concepts using frame-like structures, each having facets such as *name*, *definition*, and *examples*. The system uses some 250 heuristics (stated as condition/action rules) to guide its search through the space of concepts. These heuristics fall into three general categories – for generating new concepts, for filling in facets of existing concepts, and for determining which task on the agenda to perform next.

Lenat's system incorporates two of our proposed operators – defining composite relations and defining classes of objects. For instance, AM defines the relation of 'addition' in terms of more basic set relations, and then proceeds to define 'multiplication' as repeated addition. The system defines object classes in a model-driven way, generating a new class definition and then running experiments to determine which objects are members of that class. Thus, it defines 'even numbers' to be those 'natural numbers' that can be divided by 2, and then finds that 2, 4, 6, etc. are instances of this class.

Since AM searches a large space of relations and classes, it must restrict its attention to interesting concepts. The system uses several heuristics to this end. One of the most powerful of these rules states that if a relation has been defined in multiple ways, then it is very interesting. For example, AM's searches lead it to define multiplication in four different ways, and this in turn cause the system to devote considerable attention to this concept. Another heuristic focuses AM's processing on object classes which have neither too many nor too few elements. Thus, the system finds the class of primes quite interesting, since there are many examples of this concept, but not too many. In contrast, AM finds the class of even primes to be uninteresting, since it has only one member.

Now let us examine how AM uses these two operators to discover the concept of prime numbers. As we have mentioned, the system finds four alternative definitions for 'multiplication'. This results in a high interest value for the relation, leading AM to spend considerable time examining the concept. One of the system's many heuristics suggests defining the inverse of an interesting relation. AM applies this rule to the current concept, giving

$$\textit{divisors-of}(X,Y) \Leftarrow \textit{multiplication}(X,Y)^{-1}$$

The new relation 'divisors-of' is interesting by its association with multiplication, and AM now invokes another heuristic that suggests looking at extreme cases of interesting concepts. This leads to a number of new objects classes – numbers with zero divisors, with one divisor, with two divisors (the class of primes), and with three divisors. The first two classes turn out to have very few examples, and AM abandons them as a result. However, the system finds that there are few (but not too few) examples of numbers with two divisors and three divisors. Thus, both of these classes are considered interesting enough for further processing.

Upon closer inspection, AM finds a number of relations between these concepts. For instance, numbers with three divisors appear always to be the square of some prime number (a number with two divisors). In addition, the system also finds that every natural number can be factored into a unique set of prime numbers; this is the unique factorization theorem. It also arrives at Goldbach's conjecture that every even number is the sum of two primes. Thus, even though AM spends most of its effort in defining new object classes and relations, it also has the ability to formulate qualitative laws based on these concepts.

AM's search covers only part of the problem space we have defined, but it nevertheless has much of the flavor of an integrated discovery system. The program generates new concepts incrementally, and it designs and carries out its own experiments. It uses these experiments both to uncover qualitative relations and to test hypotheses once they have been formulated. Moreover, AM's agenda mechanism provides a sophisticated strategy for focusing attention and allocating effort. Given this sophistication, it seems surprising that more of our operators did not emerge, but this may be a function of the mathematical domain for which AM was designed.

3.2 BACON

Langley's BACON was another early machine discovery system, though it was actually a series of systems that gradually evolved over the years (Langley, 1978, 1981; Langley, Bradshaw, & Simon, 1983). The emphasis of this work was on general, weak methods for discovering quantitative empirical laws. Given a set of numeric independent and dependent terms, BACON carries out simple 'experiments' to gather data and then searches for one or more empirical laws which summarize those data. The system has discovered a variety of laws from the history of physics and chemistry, including the ideal gas law, Ohm's law for electric circuits, Snell's law of refraction, and Black's heat law. Each of these laws is represented as simple constancies or linear relations, and this is where our operators come into play. In order to state complex laws in such a simple format, the system must define terms that make this possible.

To this end, BACON uses two of our operators – defining numeric terms and postulating intrinsic properties. The system's top-level goal is to find some numeric term which has a constant value for the given data, or which is involved in a simple linear relationship. In looking for such terms, BACON carries out a depth-first search through the space of possible terms, with backtracking occurring when necessary. The program limits itself to two types of numeric terms – ratios and products – but these can be applied recursively to define more complex terms involving exponentiation.

Two main heuristics guide the search through the space of numeric terms. One of these rules notes when the values of two terms increase together; in this case, BACON defines the ratio of these terms (unless they are linearly related). Another heuristic notes when the values of one term increases as those of another decrease; in this case, the system defines the product of the two terms. Two final rules note constant values and linear relations; these do not create new terms, but instead formulate empirical laws that incorporate the terms.

For example, given the mean distance d for each solar planet along with its period p, BACON's heuristics note that the values of these terms increase together. This leads the system to define the ratio term $X = d/p$. Upon computing the values of X, BACON notes that these values increase as those of d decrease, and this causes the program to define $Y = dX = d^2/p$. When the values of Y are computed, they are found to increase as those of X decrease, leading to the product $XY = d^3/p^2$. The values of this term are nearly constant across the planets, so BACON formulates a general law that summarizes the original data. The system also includes methods for recursing to higher levels of description in order to find

laws involving multiple independent terms, but we do not have the space to discuss them here.

The need for intrinsic properties arises when BACON encounters independent terms with nominal (symbolic) values. Since the system cannot discover a numeric law from symbolic data, it is forced to 'invent' a new numeric term. The values of the intrinsic property are based on the values of the current dependent term. Thus, BACON finds a linear relation between this dependent term and the intrinsic property as soon as the latter is defined, but this relation is tautological. The system can take advantage of the new term to formulate empirically meaningful laws only when its values are used in some different context.

Let us consider an example of intrinsic properties from 18th century chemistry. When Proust began to study the quantitative aspects of reactions, he discovered that a given element always contributes the same percentage to the weight of the resulting compound. Table 2 presents some idealized data which obey Proust's law of constant proportions. For each reaction, the table lists the contributing element, the resulting compound, the weight of the element W_E, and the weight of the compound W_C.

Given these data, BACON first detects that the weight of the element increases with the weight of the compound. This leads the system to define the numeric term $X = W_E/W_C$, which has a constant value for a given element-compound pair. This ratio has a different value for different pairs of substances, but since the element and compound terms take on symbolic values, BACON cannot immediately formulate any further numeric laws. Its response is to define the intrinsic property i_p(Element, Compound) = W_E/W_C and to associate the values of this term (which are based on those of the ratio W_E/W_C) with each particular element/compound pair.* This intrinsic property corresponds to the constant weight ration discovered by Proust.

Table 2. Discovering the law of constant proportions

Element	Compound	W_E	W_C	W_E/W_C
Hydrogen	Water	10.0	90.00	0.1111
Hydrogen	Water	20.0	180.00	0.1111
Hydrogen	Water	30.0	270.00	0.1111
Hydrogen	Ammonia	10.0	56.79	0.1761
Hydrogen	Ammonia	20.0	113.58	0.1761
Hydrogen	Ammonia	30.0	170.37	0.1761

In summary, BACON relies on two of our operators – defining numeric terms and postulating intrinsic properties – and combines these operators in an effective manner. However, the system clearly searches only part of the problem space we have defined, particularly ignoring the importance of qualitative laws and the operators which support their discovery.

* This is not the best example of an intrinsic property, since it does show how such properties can contribute to non-tautological laws. However, it does convey the basic idea.

We would certainly not want to abandon the insights of BACON in future discovery systems, but these insights are certainly incomplete.

3.3 ABACUS

Unlie BACON, which discovers only quantitative relations, the ABACUS system (Falkenhainer 1985, Falkenhainer & Michalski, 1986) combines methods for quantitative and qualitative discovery. ABACUS accepts data in a similar form to those processed by BACON, though it does not require that terms be labeled as independent and dependent. From these data, the system generates numeric laws with qualitative preconditions. As a result, ABACUS can discover multiple laws which hold for different subsets of the data. For example, if a data set contains both liquid and gaseous substances along with their respective pressure, temperature, and volume, the program discovers the following relations:

IF substance = gas	THEN PV/T = *constant*
IF substance = liquid	THEN no relation found

The first of these is equivalent to the ideal gas law, with the condition that the substance be a gas stated explicitly; this version is a more cautious form that found by BACON. The second statement reveals that no analogous law holds for liquids.* Falkenhainer and Michalski (1986) present a number of examples of useful preconditions on scientific laws.

ABACUS uses one of our proposed operators, defining numeric terms, to discover quantitative relations between observable attributes. Like BACON, the system searches a space of numeric terms, looking for some term that takes on constant value; the difference is that this term need be constant for only *some* of the observations. In the example above, the numeric term $X = PV/T$ was constant for a subset of the data. In addition to products and ratios, ABACUS also defines new terms by taking sums and differences of existing terms.

The discovery system allows irrelevant variables, but these increase the size of the search space considerably and a simple BACON-like search strategy becomes ineffective. In response, ABACUS employs two new algorithms, *proportionality graph search* and *suspension search*. These search methods will converge on constant numeric terms in a reasonably efficient manner, and include the ability to handle a certain degree of noise. We will illustrate proportionality graph search as it applies to the ideal gas law.

Suppose that we extend the original data set for the discovery of the ideal gas law (the temperature T, volume V, and pressure P of a gas) to include the additional variable M. Further suppose that M is proportional to the volume V, even though this relation is irrelevant to the ideal gas law. ABACUS uses the observations to construct a *proportionality graph* like that shown in Figure 1 for the ideal gas data. The nodes of this graph represent observable variables, while a link between two nodes indicates that these two variables are either inversely or directly proportional to each other. The absence of an edge means that

* This is actually a poor example to distinguish BACON from ABACUS, since the former could actually arrive at similar laws using intrinsic properties if it were given *substance* as a nominal independent attribute. However, ABACUS can also arrive at conditional laws for cases where intrinsic properties cannot be used.

two variables are *not* related. In the figure, there is an edge between V and P because these two variables are inversely proportional to each other. There is no edge between the nodes for M and T, since there is no relation between these variables.

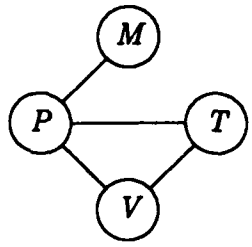


Figure 1. Proportionality Graph for Ideal Gas Law

After ABACUS has constructed this graph, it determines the largest cycle set or biconnected component. For graph in Figure 1, the largest such set is $\{P, V, T\}$. The system then focuses its attention on the members of this set in its attempt to find numeric laws, performing a depth first search with backtracking to find some new term with constant or semi-constant values. A set of heuristics similar to the one used in BACON aids this search. If the largest cycle fails to exhibit such a term, ABACUS defines a new term using variables M and P, includes this term into the set, and continues the search. Falkenhainer and Michalski argue that irrelevant variables are likely to be excluded from such cycles, so that this search will find the desired numeric term more efficiently than a simple depth first search.

Using this search method, ABACUS quickly converges on the constant numeric term $X = PV/T$, despite the presence of the irrelevant variable M. However, the authors concede that proportionality graph search encounters difficulty when complex terms (such as sums of products) are involved. If ABACUS cannot find a useful numeric term using this method, then it resorts to a second search algorithm – suspension search. This process resembles beam search but allows backtracking through the space of terms. Using this approach, the system can discover more complex laws such as conservation of momentum.

ABACUS does not explicitly use our second operator, postulating intrinsic properties, in its search for empirical laws. However, the system's use of logical preconditions leads to effects very similar to intrinsic properties. Consider again the data in Table 1, which led BACON to define the intrinsic property of combining weights and thus to Proust's law of constant proportions. Given the same data, ABACUS would note a relation between the weight of the element W_E and the the weight of the compound W_C and thus define the ratio $X = W_E/W_C$. The system would then notice that this term has semi-constant values, and would set about determining the conditions under which each value occurred. This would produce the following pair of laws:

ClassA	IF	[compound = Hydrogen]
	THEN	$W_E/W_C = 0.1111$
ClassB	IF	[compound = Ammonia]
	THEN	$W_E/W_C = 0.1761$

These state that different values of W_E/W_C are associated with different compounds.* In some sense, these associations are equivalent to those stored by BACON when it postulates intrinsic properties and infers their values. However, there are two important differences. On the one hand, BACON has the ability to retrieve its intrinsic values at some later time and incorporate them into other laws. On the other, BACON can only form intrinsic properties when the nominal variables are under experimental control, while ABACUS can form conditional expressions from observational data.

As we have seen, ABACUS combines methods for qualitative and quantitative discovery, and in this sense it approaches the type of integrated discovery system that is our ultimate goal. However, there are two quite different notions of the term 'qualitative'. Although ABACUS finds qualitative conditions on numeric laws, it does *not* discover laws involving qualitative relations such as those found by the early chemists. The system does not define classes of objects (even though its law-finding methods provide support for this activity), nor does it define composite relations or classes of such relations. Thus, like the other systems so far reviewed, ABACUS searches only a portion of the problem space that we have defined.

We should mention one further point that involves both ABACUS and BACON. As we explained earlier, the operator for defining composite objects can prove quite useful in stating laws such as conservation of momentum. Yet both ABACUS and BACON discover these laws without using this operator. The reason for this apparent inconsistency is that both systems ignore the distinction between objects and their attributes. Rather, they represent data as a conjunction of attribute-values and make no effort to associate attributes with particular objects. In the momentum case, this leads the system to view the given data – the momenta and velocities of the two colliding objects – as belonging to one 'object' rather than two separate objects. Some versions of BACON (Langley, Bradshaw, & Simon, 1982) used subscripts to aid in the search for conservation laws, but this was a weak attempt at best. We believe that future discovery systems would do well to clearly distinguish between objects and their attributes, and to form composite objects when considering a conservation law.

3.4 GLAUBER

As we have already mentioned, much of the effort in an emerging scientific discipline is devoted to classifying objects and to formulating qualitative laws. Langley, Zytkow, Simon, and Bradshaw's GLAUBER (1986) addresses both of these tasks. This system accepts as input a set of qualitative facts, such as taste(HCl, sour) and reacts({HCl NaOH} {NaCl}). GLAUBER transforms these facts into qualitative laws in which specific objects have been replaced by more abstract classes, such as 'acids' and 'alkalis'. These laws also include

* If the data had included different elements as well, ABACUS would have included these in the conditions it discovered.

universal or existential quantifiers that specify the generality of the law.

Table 3. States generated by GLAUBER in the discovery of acids, alkalis and salts

(a) Initial State

$reacts(\{HCl\ NaOH\}, \{NaCl\})$	$taste(NaNO_3,\ salty)$
$reacts(\{HCl\ KOH\}, \{KCl\})$	$taste(KNO_3,\ salty)$
$reacts(\{HNO_3\ NaOH\}, \{NaNO_3\})$	$taste(NaCl,\ salty)$
$reacts(\{HNO_3\ KOH\}, \{KNO_3\})$	$taste(KCl,\ salty)$
$taste(HCl,\ sour)$	$taste(NaOH,\ bitter)$
$taste(HNO_3,\ sour)$	$taste(KOH,\ bitter)$

(b) Intermediate State

$SALTS = \{NaCl, KCl, NaNO_3, KNO_3\}$

$\exists\, x \in SALTS\ \ni\ reacts(\{HCl\ NaOH\}, \{x\})$	$taste(HCl,\ sour)$
$\exists\, x \in SALTS\ \ni\ reacts(\{HCl\ KOH\}, \{x\})$	$taste(HNO_3,\ sour)$
$\exists\, x \in SALTS\ \ni\ reacts(\{HNO_3\ NaOH\}, \{x\})$	$taste(NaOH,\ bitter)$
$\exists\, x \in SALTS\ \ni\ reacts(\{HNO_3\ KOH\}, \{x\})$	$taste(KOH,\ bitter)$
$\forall\, x \in SALTS\ \ taste(x, salty)$	

(c) Final State

$SALTS = \{NaCl, KCl, NaNO_3, KNO_3\}$	$\forall\, x \in SALTS\ taste(x, salty)$
$ACIDS = \{HCl, HNO_3\}$	$\forall\, x \in ACIDS\ taste(x, sour)$
$ALKALIS = \{NaOH, KNOH\}$	$\forall\, x \in ALKALIS\ taste(x, bitter)$

$$\forall\, x \in ALKALIS\ \forall\, y \in ACIDS\ \exists\, z \in SALTS\ \ni\ reacts(\{x\ y\}, \{z\})$$

GLAUBER uses only one of the operators we have described in its formulation of qualitative laws – defining classes of objects. Unlike AM, which first constructs an intensional definition for some class and then generates examples, GLAUBER observes objects in the environment and classifies them based on common features and relations. For example, if a number of objects have the same taste (say sour), the system may define a new class (acids) with these objects as members. GLAUBER then generates a qualitative law which has the same form as the original facts, but in which the class name has replaced the specific objects. Such a law is guaranteed to hold for all members of the class, and so can be universally

quantified. However, GLAUBER also substitutes the class for its members in other facts, and in these cases the system must empirically determine whether a universal or existential quantifier is appropriate.

Let us consider GLAUBER's discovery of the concepts of acids, alkalis, and salts. Although the 17th century chemists did not focus on quantitative data, they had considerable qualitative knowledge of substances. This included information about the tastes of various substances, as well as the reactions in which they took part. For example, they knew that HCl had a sour taste and that this substance reacted with NaOH to form the new substance NaCl. These facts and others led the early chemists to group substances like HCl, NaOH, and NaCL into the classes of acids, alkalis, and salts.

Table 3 (a) presents a similar set of facts that were given to GLAUBER. Examining this initial knowledge base, GLAUBER notices that four of the objects (NaCl, KCl, $NaNO_3$, and KNO_3) have a salty taste, and defines a class with these four objects as members. For the sake of clarity, let us call this class 'salts'. Upon defining this class, GLAUBER replaces instances of the class with the name of the class; this substitution occurs in all facts and laws known to the system. Thus, GLAUBER adds the class SALTS = {NaCl, KCl, $NaNO_3$, KNO_3} to memory, along with the tautological law $\forall$ x $\in$ SALTS taste(x, salty). In addition, the program replaces the salts occurring in reactions with the name of this class, giving a number of more abstract reactions. However, since only one instance of each such pattern occurs, GLAUBER decides on an existential quantifier in each case. The resulting knowledge base is shown in Table 3 (b).

At this point, GLAUBER proceeds to define the class $ACIDS = \{HCl, HNO_3\}$, based on the observation that both HCl and HNO_3 have sour tastes. This time, after substitution occurs, the system decides that universal quantification is justified for the reaction laws and it proposes two general laws:

$$\forall \text{ x} \in \text{ACIDS } \exists \text{ y} \in \text{SALTS } \ni \text{ reacts}(\{\text{x NaOH}\}, \{\text{y}\})$$

$$\forall \text{ x} \in \text{ACIDS } \exists \text{ y} \in \text{SALTS } \ni \text{ reacts}(\{\text{x KOH}\}, \{\text{y}\})$$

However, these new laws have identical forms, leading GLAUBER to define a third class of substances, ALKALIS = {NaOH, KOH}. This results in the general reaction law shown in Table 3 (c), along with another law describing the taste of alkalis. At this point, the system has successfully summarized all of the original data, so it halts with three classes and four qualitative laws.

Jones (1986) has described NGLAUBER, a successor to GLAUBER that improves on many aspects of the initial system. For instance, GLAUBER required all data to be present at the outset, while NGLAUBER processes data incrementally. In addition, Jones' system is able to distinguish between unobserved facts and disconfirming evidence, such as missing and failed reactions. Although the two systems employ the same operator for defining object classes and formulate similar laws, NGLAUBER uses quite different heuristics than its predecessor. The earlier program operated nonincrementally because it relied on frequency information to decide which classes to form. In contrast, NGLAUBER forms whichever classes are suggested by the most recent data it has examined, but has the ability to backtrack

if these classes predict disconfirming evidence. This seems a more plausible model of human scientists than does Langley et al.'s system.

Although GLAUBER and NGLAUBER employ only one of the operators that underly empirical discovery, they fill an interesting niche nonetheless. They show that data-driven heuristics can be used to propose useful classes. They also suggest that some classes are best characterized not by independent features, but by relations between the classes themselves. Finally, the systems point out the need for distinguishing between universal and existential quantification in qualitative empirical laws. We believe that all of these features should be kept in mind in designing more complete, integrated discovery systems.

3.5 OPUS

Another important form of empirical discovery is known as *conceptual clustering*. Basically, this is the task of taxonomy formation, with the added constraint that one formulate an intensional description for each class in the resulting conceptual hierarchy. Since Michalski and Stepp (1983) first defined this problem, a number of conceptual clustering systems have been developed and tested. Rather than attempting to review all of these programs in an already lengthy paper, we will focus on Nordhausen's (1986) recent OPUS system, which has a number of features that are interesting from our perspective.

As we have seen, objects can be described not only in terms of independent attributes, but also through their relation to other objects. OPUS uses both kinds of information to formulate new classes and to find qualitative laws describing those classes. OPUS inputs a set of objects described by nominal attributes such as *color* and *size*, along with binary relations between objects, such as *eat* or *parent*. From these data, the system produces a hierarchical classification tree along with a concept description which uniquely identifies each class.

In constructing this taxonomy, OPUS uses two of the operators we have proposed – defining new classes and defining composite relations. The system defines composite relations in terms of existing relations and simple attributes such as color or size. For example, it combines the binary relation *offspring(X,Y)* and the attribute *color(X,c)* to define the composite relation

$$\textit{offspring-color}(X,c) \Leftarrow \textit{offspring}(X,Y) \;\&\; \textit{color}(Y,c)$$

Once OPUS has defined composite relations, it uses them as attributes during the process of defining object classes. For instance, *offspring-color* can be used to distinguish peas which have only yellow offspring and peas which have both green and yellow offspring. OPUS classifies objects using both primitive attributes (such as color) and attributes that have been derived from relations.

OPUS builds its classification tree in a top-down manner. At each branch the system divides objects into mutually exclusive subclasses, with members having some value of an attribute in common. For example, if the attribute 'color' is used to partition objects, OPUS divides the objects into classes with members of the same color. The program then selects

that attribute which best divides the current object set according to two criteria. The *simplicity* criterion favors classes with simple descriptions, while the *inter-cluster difference* criterion promotes classes with different properties. If none of the existing attributes can distinguish between the existing set of objects (i.e., if members of all classes have the same value for the given attributes), then OPUS defines new attributes and uses these to define new classes. This process is recursive, so that defined attributes can be used as the basis for more complex attributes.

Now that we have described OPUS in the abstract, let us examine its use of the two operators in rediscovering the classes of hybrids and purebreds from the early days of genetics. In this domain, OPUS is provided with information about the color of various peas (green or yellow), along with the parent-child relations between different peas. For example, pea A might be described as $color(A, green)$ and $parent(A, B)$. At the outset, OPUS uses the primitive attribute color to define the classes of yellow peas and green peas. But because no distinctions can be made on the basis of existing attributes, the system defines two composite relations for this purpose: *offspring-color(X,c)* and *parent-color(X,c)*.

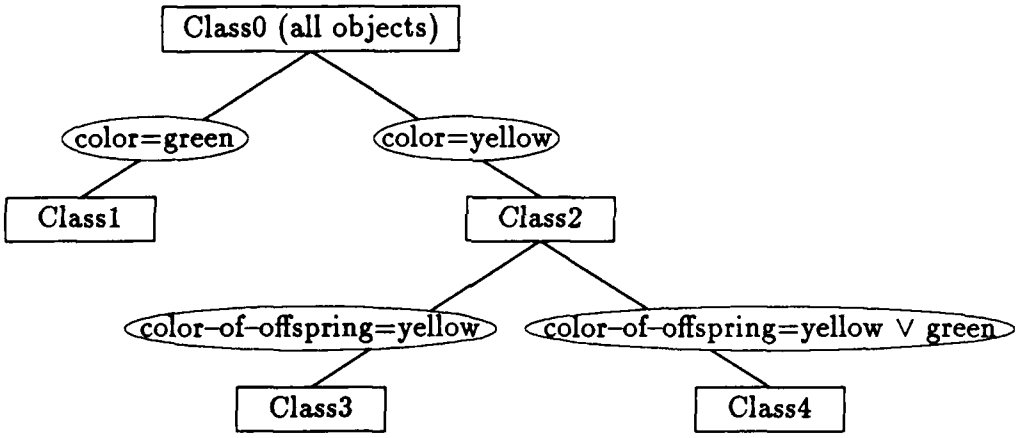


Figure 2. Classification tree equivalent to Mendel's definitions

Both relations can then be used as attributes to refine the existing classes. In this case, the attribute *offspring-color* does a better job of partitioning the objects, so OPUS selects this term to extend the classification tree. As a result, the system refines the class of yellow peas into two subclasses – those which produces only yellow offspring and those which produce both yellow and green offspring. At this point, OPUS has not only formulated the classes of hybrids and purebreds; it has also described these classes using concepts very similar to the ones proposed by Mendel.

Elements of the class of purebreds have purebred offspring.
Elements of the class of hybrids have purebred and hybrid offspring.

OPUS continues this process, further refining the purebred class into those with hybrids as parents and those with purebreds as parents. Figure 2 presents the final taxonomy generated by the system; this is very similar to the organization proposed by Mendel in the 1860's.

OPUS is interesting along a number of dimensions relevant to our framework. Like AM, this system defines both object classes and new relational terms. However, it applies these operators in quite different contexts and to quite different ends than did Lenat's early system. Nor is OPUS a traditional conceptual clustering system, since it focuses on relations between objects as well as isolated features of those objects. But the most interesting aspect of the system lies in the interaction between the two operators. OPUS defines composite relations in order to support the creation of new object classes, just as BACON postulates intrinsic properties in order to allow the creation of useful numeric terms. This is precisely the type of interaction we would hope for in an integrated system, in which each of the six operators feed off the results of the others to create powerful synergies that aid the discovery process.

3.6 Summary

In this section, we reviewed five existing empirical discovery systems in the light of our framework. We summarize the results of this analysis in Table 4. Cells marked with crosses indicate operators that clearly exist within the specified system, while triangles indicate ambiguous cases where the operator is absent, but where the system achieves a similar effect indirectly. The most obvious characteristic of the table is its sparsity; very few of the possible cells are occupied. In fact, none of the systems incorporate more than three of the operators, even with a liberal interpretation.

Table 4. Discovery systems and their operators

System	*numeric term*	*intrinsic property*	*composite object*	*class of objects*	*composite relation*	*class of relations*
AM				×	×	
BACON	×	×	△			
ABACUS	×	△	△			
GLAUBER				×		
OPUS				×	×	

This means that each of these AI discovery systems search only a portion of the problem space of defined terms that we described earlier, and this limits the class of laws that each system can discover. This in turn suggests a natural goal for future research – the design and construction of an integrated discovery system that employs all six operators to search the entire problem space. In the following pages, we describe our plans for such a system.

4. An Integrated Model of Empirical Discovery

Although we believe our framework for empirical discovery has helped to clarify and unify earlier work in the area, it has only limited usefulness. Our ultimate goal is to translate this framework into an integrated system discovery system. Only by following this path can we determine whether our operators are necessary and sufficient for empirical discovery, and identify heuristics to direct the application of these operators in an intelligent fashion. In this section, we detail our plans for an integrated discovery system (IDS) that incorporates all six of the operators we have proposed. However, in order to realistically simulate the discovery process, one needs some environment which is separate from the discovery system, but which that system can inspect and manipulate. We are implementing such an environment for the domains of early physics and chemistry which obeys the major laws of these domains. Below we describe the environment in some detail, before turning to our designs for the discovery system itself.

4.1 Objects and Attributes

The simulated environment contains a set of *objects*, each having a variety of attributes. These attributes are similar to those available to early physicists and chemists, such as volume, color, taste, shape, location, temperature, and mass. Many of these attributes are numeric in nature, but others (like color and taste) are usually viewed as nominal (symbolic). However, we have also chosen to represent these as numeric terms with real values, since we feel this more closely reflects the situation encountered by the early scientists. Thus, the taste of an object involves three sub-attributes – saltiness, sourness, and bitterness – each taking values from zero to one. We use similar sub-attributes to represent the colors of objects.

A few attributes seem genuinely nominal, at least for our purposes. For instance, the *state* of an object can be *solid*, *liquid*, or *gaseous*. These values represent qualitatively different aspects that one can determine through direct inspection. Similarly, the *shape* of an object takes on the nominal values *box*, *sphere*, *cylinder*, or *irregular*. Although these certainly do not exhaust the possible shapes occurring in the physical world, they provide enough variety to allow interesting behavior.

In addition, primitive objects can be connected to form more complex composite objects.* Thus, one can specify that two or more primitive objects are *parts* of a complex object. These components must move together and are affected together along other dimensions (such as temperature). The environment supports three forms of object composition. *Generic* composition simply specifies that two objects are part of a composite object, but the two other forms specify additional features. Composition by *containment* specifies that one object is contained by another. This is essential if our system is to replicate early chemical discoveries involving gases and liquids. Similarly, two containers may be *connected* by a conduit, allowing the contents to move from one object to the other. These relations let

* We are talking here about the physical combination of objects. The reader should not confuse this with our third operator, which involves the *logical* composition of objects.

one construct reasonably complex systems of objects. Finally, two objects can *touch* one another; this relation does not define a composite object, but many laws include adjacency as an application condition.

An important aspect of the environment is that it changes over time. Thus, the temperature of object A at one instant may differ from its temperature at the next instant. Some attributes may well have constant values, but this is something the system must discover for itself. In other cases, the system must formulate laws that describe an object's change over time. In addition, new objects may enter the world and existing objects may disappear (as in chemical reactions). The discovery system must be able to summarize these qualitative changes as well as quantitative ones. These possibilities will force us to handle laws and explanations of a quite different nature than those we addressed in previous research.

4.2 Gathering Data and Performing Experiments

The discovery system will observe the world through a set of *sensors*. These are passive in nature, simply letting the system inspect the value of an object along a certain dimension; they correspond to primitive measuring instruments, such as rulers, scales, and thermometers. In general, one sensor exists for each observable attribute. Thus, at any given time, the system can measure the following properties of any given object: mass, temperature, color (lightness, hue, saturation), taste (saltiness, sourness, bitterness), location (x and y coordinates), size (radius; length, width, depth), texture, shape, and state.

Some sensors can be applied only to certain objects. For instance, the system can inspect the radius of spherical objects and the length, width, and depth of boxes, and from this one can easily compute their volumes. However, one cannot directly measure the dimensions of irregular objects, and this makes the derivation of volume more difficult. Restrictions also apply to the components of complex objects. The system can measure the color, temperature, and locations of the components independently, but it cannot directly measure these values for the composite object. On the other hand, it can measure the mass of composite objects, but not the mass of their components. The system may be able to infer these values, but this requires intelligent behavior rather than simple sensing.

Most earlier discovery systems were provided with data, but in this environment one must actively gather information. If the system wants to measure the mass of object A during some time cycle, it must explicitly call on its mass sensor with A as the argument. Moreover, the number of such measurements that can be made during a given cycle is limited.* Thus, the system must focus its attention on objects and aspects of those objects that it decides are important.

In addition to sensors, the simulated environment also supports active processes called *effectors*. These let one affect objects directly, including actions such as changing the location of an object, breaking an object into two equal components of the same type, and heating an object. Like sensors, the effectors require an intentional act on the part of the system.

---

* We plan to start by allowing 10 sensors to be applied simultaneously, but we may reduce or increase this limit based on our experience.

These actions also let the system construct composite objects using the composition relations (generic, containment, and connection) described earlier. Thus, one can construct simple experimental configurations by rearranging objects and their relations to each other.

More important, one can run simple experiments by creating initial conditions and then using sensors to observe changes over time. For instance, one might place two objects in contact and heat them both. In some cases, a new object with different features will be created, and the mass of this object will increase over time as the masses of the original objects decreases. In this way, we can simulate simple chemical reactions.

Note that such experiments provide a way of defining new measuring instruments. Thus, one might measure the volume of liquid held in some container, place an irregular object in the container as well, and then measure the resulting volume. Archimedes used a similar strategy to measure the volumes of irregular objects, and this ability provides an interesting range of behaviors that have been largely ignored in work on machine discovery. Another example of a new 'measuring instrument' involves sensing the temperature of an object, heating it at constant rate for some time, and resensing the temperature. Together with the elapsed time, these temperatures let one estimate the specific heat of the object.

Now that we have described the environment in which our discovery system (IDS) will operate and the primitive actions it has available for interfacing with that environment, let us turn to the system itself. We have divided our discussion into two parts, the first dealing with qualitative discovery and the second handling the formulation of quantitative laws. We will see that all six of our operators are embedded within the design of IDS, and that the system's methods for numeric discovery build naturally upon the qualitative laws it constructs at the outset.

4.3 Inferring Qualitative Schemas from Behavior

Before it can discover numeric relations, our discovery system must first determine the basic types of events that occur in its surroundings. The system will begin by examining individual objects, looking for terms that are constant over time. Most attributes of objects will be constant over time until some effectors are applied. Based on these constancies, the system will generate an initial taxonomy, grouping similar objects together. This activity corresponds to the operator for defining classes of objects. The first such classes will be chemical substances, the members of which have the same color, texture, taste, and density (a defined term), but which have different masses and volumes. More abstract classes such as metals (which are smooth and shiny) and acids (which taste sour) may also be defined, but members of these groups will have fewer features in common.

Once an initial set of classes have been identified in this manner, the system will use them in designing experiments and in generalizing the results of those experiments. This involves applying effectors to members of different groups and observing the results. Let us consider a simple experiment as an example. Suppose one fills container C_1 with liquid L_1 to height H_1 and fills container C_2 with liquid L_2 (of the same class) to height H_2, and then connects these two containers with an open conduit. As time passes, one observes the heights of liquid in each container, noting that one level increases and the other decreases

until the two levels are equal, having reached equilibrium.

If we focus on the qualitative aspects of this situation, only two classes of states exist. The first class can be described by three relations: $L_1 > L_2$, $\Delta L_1 < 0$, and $\Delta L_2 > 0$. Similarly, the second (equilibrium) class of states can be described by different relations: $L_1 = L_2$, $\Delta L_1 = 0$, and $\Delta L_2 = 0$. These classes can be easily induced from the manner in which the system changes over time. Moreover, one can also infer that the first class of states leads to the second class; in other words, any non-equilibrium situation is gradually transformed into an equilibrium situation. We represent this qualitative schema graphically in Figure 3; the illustration includes the alternative situation, in which one begins with $L_1 < L_2$.

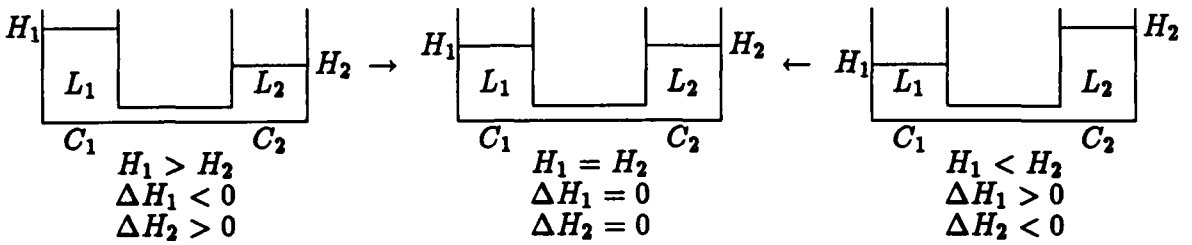


Figure 3. Qualitative schema for fluid flow

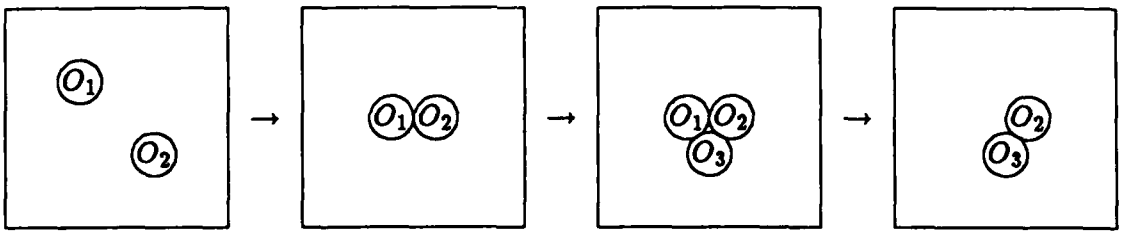
The representation we have used for this qualitative schema is very similar to that proposed by Forbus (1984) in his qualitative process (QP) theory. However, note that we have no model of the *processes* responsible for the transition between states in our schema. Rather than inferring the schema from process knowledge (as Forbus does with his envisionment mechanism), IDS will induce the schema by observing changes in the environment over time. In some sense, our schema represents process knowledge in its own right, but uses a form quite different from that used in QP theory.

Now let us consider a more complex example involving a chemical reaction. Suppose we move two objects O_1 and O_2 into contact with each other, and that a new object O_3 is generated as a result. Moreover, imagine that the masses of O_1 and O_2 decrease over time until O_1 reaches zero (and thus disappears), while the mass of O_3 increases in the meantime. Finally, suppose the reaction ends with the masses of O_2 and O_3 remaining constant over time.

As before, we can represent these changes with a qualitative schema like the one shown in Figure 4. The first box shows the initial class of states during which O_1 and O_2 are being moved closer together. Letting D be the distance between two objects and M be the mass of an object, a number of change relations hold during these states: $\Delta D(O_1, O_2) < 0$. $\Delta M(O_1) = 0$, and $\Delta M(O_2) = 0$. Note that we include terms with constant derivatives, provided these derivatives change elsewhere in the schema. After the two objects have been brought together, the new relation $\Delta D(O_1, O_2) = 0$ replaces $\Delta D(O_1, O_2) < 0$, since the relative positions of the objects are constant.

The transition from the second class of states to the third class introduces the new object O_3. We believe that the creation or destruction of an object is always sufficient justification

for establishing state boundaries. Moreover, the qualitative relations have changed again. During this class of states, the distances between objects remain the constant zero, but the masses change: $\Delta M(O_1) < 0$, $\Delta M(O_2) < 0$, and $\Delta M(O_3) > 0$. In the transition to the final state-class, the object O_1 is destroyed, and the masses of O_2 and O_3 remain constant during these states. Taken together, these successive state descriptions form a qualitative description of the events that occur during a simple chemical reaction.



State 1	State 2	State 3	State 4
$D(O_1, O_2) > 0$	$D(O_1, D_2) = 0$	$D(O_1, D_2) = 0$	
$\Delta D(O_1, O_2) > 0$	$\Delta D(O_1, O_2) = 0$	$\Delta D(O_1, O_2) = 0$	$M(O_1) = 0$
$\Delta M(O_1) = 0$	$\Delta M(O_1) = 0$	$\Delta M(O_1) < 0$	$\Delta M(O_2) = 0$
$\Delta M(O_2) = 0$	$\Delta M(O_2) = 0$	$\Delta M(O_2) < 0$	$\Delta M(O_3) = 0$
		$\Delta M(O_3) > 0$	

Figure 4. State descriptions for chemical reaction

Although IDS will form such qualitative schemas on the basis of a single experiment, note that the resulting description is quite general. In fact, one can view the above process as defining a composite relation; this is one of the six operators we discussed earlier.* Thus, IDS might use the name *reacts* to refer to the qualitative schema in Figure 4, and specify a successful instantiation of the schema involving objects O_6, O_7, and O_9 as *reacts*(O_6, O_7, O_9). Such a representation could be passed directly to a GLAUBER-like subroutine, which would define new classes of objects and formulate qualitative laws.

Of course, one must still carefully select the objects used in the experiments to maximize the likelihood of useful results. However, recall that IDS will have already grouped objects into initial classes based on common features, and it can use these classes to constrain the process of experimentation. For instance, the system might decide to combine members of the class of sour-tasting objects (acids) with each other, but no reaction would occur in these cases and it would give up after a few unsuccessful attempts. However, the system would have more success when combining acids with members of the bitter-tasting class (alkalis). Moreover, the outputs of these reactions (salts) may never have been observed before, giving IDS a new class of objects to use in other experiments.

---

* In some sense, the generality of these schemas makes them *classes* of relations. Rather than starting with specific schemas and forming more general ones, we envision IDS as starting with very general relations which share the same qualitative descriptions. The system would then gradually form more specific versions of these schemas that differ in their quantitative features.

4.4 Finding Quantitative Laws

Once a qualitative schema has been formulated, it provides the *context* within which numeric laws can be framed. One of BACON's drawbacks was that it failed to specify the situations under which its quantitative laws held, and IDS's qualitative schemas provide a formalism for doing this. In particular, each of the qualitative relations that occur in the schema may be transformed into a quantitative law, which is then attached to that class of states. For instance, in our equilibrium example we found that the level of one liquid decreased as the level of the other decreased. A numeric law might specify the exact rates at which these changes occurred. Another numeric law might state the final level of equilibrium as a function of the initial levels of the liquids.

Thus, IDS would repeat the same 'experiment' with different numeric parameters, instantiating the same qualitative schema in different ways. In the equilibrium example, the system could fill the containers to different initial levels and observe the resulting rates of change and equilibrium states. In the chemical reaction example, it could not use the same objects, since these are transformed during the reaction, but it could use the same classes of objects (such as *ammonia* and *sulfuric acid*). In this case, it would vary the initial masses involved in the reaction and observe the masses remaining afterwards.

We envision IDS using BACON-like heuristics to direct the search for numeric laws. The system would consider the product of two terms if they increase together and consider their ratio if one increases as the other decreases. Our experience with BACON suggests that such heuristics are quite robust even in the presence of significant noise, provided the laws involve only a few parameters. In addition, once IDS has discovered a numeric law for one object/class or pair of objects/classes, it will predict that the same law will hold for other objects/classes, even though the numeric parameters differ. When this occurs, the system will associate each value with the object or class, storing it as an intrinsic value that may be retrieved in other situations as well. Thus, IDS will include two more of the operators for empirical discovery – defining numeric terms and postulating intrinsic properties.

Let us consider the example involving chemical reactions in more detail. Suppose IDS places an object from the nitrogen class into contact with another object from the oxygen class, and that the object which emerges from the reaction has features of the nitric oxide class. Further, suppose the system runs this same basic experiment with different amounts of nitrogen (say 1.0 gram, 2.0 grams, and 3.0 grams) while holding the amount of oxygen constant at 6.0 grams. Each of these experiments will obey the qualitative schema shown in Figure 4, with object O_3 (nitric oxide) being created and object O_1 (nitrogen) disappearing.

Upon examination, IDS would find varying amounts of oxygen in each case (4.86 grams, 3.72 grams, and 2.58 grams). Comparing these values to the masses of nitrogen used in each case, it would note a linear relation with slope -1.14 and an intercept of 6.0. Varying the initial amount of oxygen causes the intercept to vary, but the slope remains constant at -1.14. This constant term corresponds to the *combining weight* of oxygen with respect to nitrogen when these two chemicals combine to form nitric oxide. Based on this constancy, the system would define an intrinsic property and associate this particular value with the

nitrogen-oxygen-nitric oxide triple.* Different intrinsic values for this term would be found for other chemical reactions that obeyed the same qualitative schema.

Taken together, the linear relation and intrinsic property specify a numeric law that describes the quantitative behavior of the schema in Figure 4. This law relates the $M(O_2)$ term occurring in the final class of states to the $M(O_1)$ term occurring in the original state-class. The IDS system would discover similar laws relating the final value for $M(O_1)$ (when this object remains) to the initial value of $M(O_2)$, and relating the final value for $M(O_3)$ to the initial values for $M(O_1)$ and $M(O_2)$.** These laws correspond to Proust's law of constant proportions. We have not considered the changes that occur in volume along with changes in mass, but if IDS focused on this term as well, it would also arrive at Gay-Lussac's law of combining volumes.

Although BACON rediscovered both Proust's and Gay-Lussac's laws, it did so in a much different form than just described. Both its data and its laws were stated in very abstract terms, divorced from any description of the physical situation involved. In the new framework, the data consist of instantiations of the given qualitative schema, and the laws relate numeric terms that occur in that schema. In addition to providing a context for numeric laws, such schemas also make possible a new class of relations that BACON did not consider – laws describing rates of change. Since the initial qualitative relations are described in terms of derivatives, it seems natural for the quantitative component of IDS to identify the constants associated with these derivatives, and (if they exist) to store them as intrinsic properties of the objects or classes involved in the reaction. We plan to explore methods for discovering such laws as well, though we have not yet formulated the details.

4.5 Summary

In this section, we outlined our plans for IDS, an integrated discovery system that instantiates the framework we proposed earlier in the paper. The system will interact with a simulated physical world through a set of sensors and effectors, and these will let IDS implement simple experiments and design new measuring instruments. In addition, the environment will change over time, forcing IDS to represent and discover types of laws that earlier machine discovery systems have ignored. The program will focus first on defining useful classes of objects, as well as determining qualitative schemas that describe changes over time. Once these schemas have been established, they will provide the context for discovering numeric laws.

Although our concern here has been with empirical discovery, IDS's schema representation also suggests an approach to theory formation. We have focused on empirical laws that deal with macroscopic events in which one can directly observe objects and changes

---

* Rather, it would define a composite object with nitrogen, oxygen, and nitric oxide as components, and associate the intrinsic value with this new object. This constitutes another of our six operators.

** In fact, the procedure of combining two objects through a chemical reaction and measuring the slope of the line relating their masses can be viewed as a new, higher level sensor for measuring combining weights. In some sense, the system will have defined a new measuring instrument.

in those objects. However, much of scientific discovery involves formulating explanations of laws and behavior in terms of structures and events that *cannot* be observed. The caloric theory and the kinetic theory of gases are two well-known examples of such explanations. Basically, we believe that explanatory theories can be formed through a process of analogy with schemas based on macroscopic phenomena. These analogies are cued by similar qualitative changes, and lead one to infer physical structure (such as the coloric fluid) that are not directly observable.

We do not have the space to consider this process in detail, and our ideas on theory formation are still rather vague in any case. But we find it encouraging that the notion of a qualitative schema may prove useful in theory formation as well as during the discovery of empirical laws. This suggests that our design for IDS will prove a fertile one for modeling the process of discovery.

5. Conclusions

Scientific discovery is a complex phenomenon involving many interacting components. Even the process of empirical discovery is sufficiently complex that earlier research on machine discovery has addressed only parts of the overall task. In this paper, we presented a general framework for empirical discovery that we hope will further our understanding of this process. Like much of the work in AI and machine learning, our framework is based upon the notion of a problem space, and we have spent much of the paper describing the operators that define that space. But rather than focusing on operators for law discovery per se, as one might expect, we focused instead on operators for defining new terms. There is ample precedent for this, since the existing machine discovery systems spend more effort in finding useful terms than they do in finding empirical laws.

We proposed six types of terms that prove useful in empirical discovery, each with an associated operator responsible for its definition. We attempted to justify each of these types with examples from the history of science, and we also used historical data to suggest a possible ordering on the operators. We found that all but one of the operators had been used in existing machine discovery systems, but that none of these systems employed more than three of the operators. In other words, previous research on machine discovery has limited itself to small portions of the total problem space. This has been a useful strategy, but we feel the time has come to construct an integrated discovery system that explores the entire space of terms and thus discovers a much wider range of laws.

In fostering this effort, we have constructed a simulated environment with which our integrated system (IDS) will interact. The system will have sensors for measuring directly observable attributes of objects, as well as effectors for running simple experiments. Objects in the environment will change over time, introducing a factor that has been absent from earlier AI work on discovery. Within this framework, IDS will begin by constructing qualitative schemas (composite relations) that summarize changes over time. The system will run experiments to determine which objects obey these schemas, and this in turn will lead to classes of objects and relations.

Once such a qualitative schema is well understood, IDS will attempt to determine the

quantitative laws that govern that schema. This will lead the system to define numeric terms, intrinsic properties, and composite objects. Moreover, the schema will provide a context within which such numeric laws can be interpreted; this is quite different from the abstract quantitative relations formulated by BACON and ABACUS. Finally, we have plans to move beyond empirical discovery and into the realm of explanation, using the same representation of events for empirical laws and scientific theories.

We believe this approach will lead to a robust and integrated system for empirical discovery, but our work on this system is still in the planning stages. The most important part of the effort remains; we must translate our ideas into a running program, and we must test this system on a wide range of discovery tasks to ensure its power and generality. However, we believe that our framework for empirical discovery has already proved useful in both clarifying earlier work in the area and in proposing directions for more powerful systems. But the approach we are taking with IDS is not the only instantiation of this framework. We encourage our colleagues to develop other approaches to empirical discovery that explore the same problem space using different methods. Working together, we can achieve both a broader and a deeper understanding of the complex phenomenon called 'discovery'.

Acknowledgements

We would like to thank Randy Jones, Don Rose, and John Gennari for their comments on an early draft of this paper and for discussions that led to the ideas presented therein. This research was supported by Contract N00014-84-K-0345 from the Information Sciences Division, Office of Naval Research.

References

Arons, A. B., *The Development of Concepts of Physics: The Rationalization of Mechanics to the First Theory of Atomic Structure*, Addison–Wesley, Reading, MA, 1965.

Bradshaw, G. L., Langley, P., and Simon, H. A., "Studying Scientific Discovery by Computer Simulation," *Science*, Vol. 222, No. 4267, Dec. 1983.

Dampier, W. C., *A History of Science*, MacMillian, New York, 1943.

Falkenhainer, B. C., "Proportionality Graphs, Units Analysis, and Domain Constraints: Improving the Power and Efficiency of the Scientific Discovery Process," *Proceedings of the Eight IJCAI*, Los Angeles, pp. 552–554, 1985.

Falkenhainer, B. C. and Michalski, R. S., "Integrating Quantitative and Qualitative Discovery: The ABACUS System." To appear in *Machine Learning* 1986.

Forbus, K. D., "Qualitative Process Theory," Technical Report No. 789, AI Lab, MIT, 1984.

Forbus, K. D., "Interpreting Observation of Physical Systems," Report No. UIUCDCS–R–86–1248, University of Illinois at Urbana–Champaign, 1986.

Jones, R., "Generating Predictions to Aid the Scientific Discovery Process." To appear in *Proceedings of AAAI–86*, Philadelphia, PA, 1986.

Laird, J. E., Rosenbloom, P.S. and Newell, A., "Towards Chunking as a General Learning Mechanism," *Proceedings of AAAI-84*, Austin, TX, pp. 188–192, 1984.

Langley, P., "BACON.1: A General Discovery System," *Proceedings of the Canadian Society for Computational Society of Intelligence*, Toronto, pp. 173–180, 1978.

Langley, P., "Data–Driven Discovery of Physical Laws," *Cognitive Science*, Vol. 5, pp. 31–54, 1981.

Langley, P., Bradshaw, G. L. and Simon, H. A., "BACON.5: The Discovery of Conservation Laws," *Proceedings of the Seventh IJCAI*, Vancover, B.C., pp. 121–126, 1981.

Langley, P., "Data–Driven and Expectation Driven Discovery of Empirical Laws," *Proceedings of the Canadian Society for Computational Society of Intelligence*, Saskatoon, Saskatschewan, pp. 137–143, 1982.

Langley, P., Bradshaw, G. L. and Simon, H. A., "Rediscovering Chemistry with the BACON System," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., Tioga Press, Palo Alto, CA, pp. 307–329, 1983.

Langley, P., Bradshaw, G. L., Zytkow, J. M., and Simon, H. A., "Three Facets of Scientific Discovery," *Proceedings of the Eighth IJCAI*, Karlsruhe, W. Ger., pp. 465–468, 1983.

Langley, P., Simon, H. A., Zytkow, J. M., and Fisher, D. H., "Discovering Qualitative Laws," Technical Report 85–15, Department of Information and Computer Science, University of California, Irvine, 1985.

Langley, P., Zytkow, J., Simon, H. A., and Bradshaw, G. L., "The Search for Regularity: Four Aspects of Scientific Discovery," in *Machine Learning: An Artificial Intelligence Approach, Vol. II*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (Eds.), Morgan Kaufmann, Los Altos, CA, pp. 425–469, 1986.

Leicester, H. M. and Klickstein, H. S., *A Source Book in Chemistry*, Harvard University Press, Cambridge, MA, 1963.

Lenat, D. B., "Automated Theory Formation in Mathematics," *Proceedings of the Fifth IJCAI*, Cambridge, MA, pp. 833–841, 1977.

Lenat, D. B., "The Ubiquity of Discovery," *Artificial Intelligence*, Vol. 9, No. 3, pp. 257–285, 1978.

Lenat, D .B., "AM: Discovery as Heuristic Search," in *Knowledge–Based Systems in Artificial Intelligence*, R. Davis and D. B. Lenat, McGraw–Hill, New York, 1982.

Lenat, D. B. and Brown, J. S., "Why AM and EURISKO Appear to Work," *Artificial Intelligence*, Vol. 23, No. 3, pp. 269–294, 1984.

Magie, W. F., *A Source Book in Physics*, Harvard University Press, Cambridge, MA, 1963.

Michalski, R. S., "Knowledge acquisition through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts," *International Journal of Policy Analysis and Information Systems*, Vol. 4, pp. 219–243, 1980.

Michalski, R. S. and Stepp, R. E., "Learning from observation: Conceptual clustering," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., Tioga Press, Palo Alto, CA, pp. 331–363, 1983.

Neves, D. M. and Anderson, J. R., "Knowledge Compilation Mechanism for the Automatization of Cognitive Skills," in *Cognitive Skills and Their Aquisition*, Lawrence Erlbaum Associates, Hillsdale, N. J., pp. 57–84, 1981.

Nordhausen, B., "Conceptual Clustering Using Relational Information." To appear in *Proceedings of AAAI–86*, Philadelphia, PA, 1986.

Singer, C., *A Short History of Science to the Nineteenth Century*, Clarendon Press, Oxford, 1941.

END

11-86

DTIC